RELATION ALGEBRA GUIDE:

Folder relation algebra contains following files.
1. main.out : Executable file
2. mainfile.cpp : source code file
3. instructor.csv : input relation file
4. students.csv : input relation file
5. queries.txt : file containing demo queries
6. Readme.pdf : This file

DEMO TEST:

1. In the terminal run main.out executable file.
2. Your screen should display some refrence instructions, data from both the csv files will appear in tabular format and your screen will be prompted with “>>” symbol where you can type in your queries. EXAMPLE : >> TYPE YOUR QUERY HERE.
3. Copy reference query from reference instructions and paste after the “>>” symbol and hit enter.You will see the output of your query in tabular format in “result” realtion.
4. You can type other queries or you can exit program by typing “exit”.

RELATIONS USED:

We use .csv files to serve the purpose of relations.In queries you can refrence instructor relation using “instructor” and student relation using “student”.You are allowed to use only these two relations.

ALLOWED OPERATIONS

1. SELECT ($) : you can perform select operation with “$” with assosciated conditions specified in “[ ]” where conditions are joined using “|” (or) or “^” (and) followed by relation name.
   Example: $[DEPARTMENT=’CSE’^ROLL>=30](student)
   Note : In a query you can join several conditions using either “^” or “| “ at a time but not both.

2. PROJECT(@) : You can display selected attributes from a relation using “@” with assosciated attributes specified in “[ ]” where different attribues are seperated by “,”.
   Example :@[DEPARTMENT, NAME](student)

3. RENAME(#) : you can rename a relation and its attributes using “#” and you can specify the new name of relation in “[ ]” with the new names of attributes in “{ }”.
   Example : #[x{Sname,Sroll,Sdept}](student).

4. Cartesian Product(*) : you can perform cartesian product between two relations using “*” . Both realtions should have different names as well as different attribute names.
   Example : (student * instructor).

5. Set Difference(-) : you can perform set difference between two realtions using “-”
   You can have only one set difference operator in a query.
   Example : [NAME](student) - [INSTRUCTOR](instructor)

6. Set Union (+) : You can perform union of two relations using “+”.
   you can have only one set union operator in a query.
   Example : [NAME](student) +[INSTRUCTOR](instructor).

NESTED QUERIES

You can perform nested queries and obtain more useful results.You can perform project, select rename upto any depth but set difference and set union are allowed once per query.
Example 1: project + select
@[NAME,ROLL]($[ROLL>=130^DEPARTMENT='CSE'](student))

Example 2: select + project + rename
@[STUD_NAME,STUD_DEPT]($[DEPARTMENT='CSE'](#[x{STUD_NAME,STUD_ROLL,STUD_DEPT}](student)))

Example 3: Cartesian product+ select+ project+set difference
@[NAME]($[DEPARTMENT=DEPT](student*instructor))-@[NAME]($[SECTION=TEACHES_SEC](student*instructor))

MAINFILE.CPP

All the source code is contained in mainfile.cpp. All the code has been commented for better readability .Comments above the functions describe what the function does and all the main steps which the function performs.Also inside the functions we describe what main lines of code are doing.

QUERIES.TXT

This text file contains some demo valid queries for testing purposes and also some invalid queries to show how errors are handled by the code.